提供了过滤虚假评价的方法：(应对bad-mouth ballot-mouth)
通过 α 和 β 得到 证据个数. Beta分布的假设！
1)confidence(和个数有关)！！！，2)deviation(偏差太大的过滤掉//评价偏差 和 confidence偏差)，3)位置上的距离。
=>通过Cluster算法 得到过滤后的集合
最后加权平均得到最后的评价

Recommendation Based Trust Model with an Effective Defence Scheme for MANETs

Antesar M. Shabut, Keshav P. Dahal, *Senior Member, IEEE*, Sanat Kumar Bista, and Irfan U. Awan

ad hoc里的trust ！！！！
有可能包含不错的参考文献

Abstract—The reliability of delivering packets through multi-hop intermediate nodes is a significant issue in the mobile ad hoc networks (MANETs). The distributed mobile nodes establish connections to form the MANET, which may include selfish and misbehaving nodes. Recommendation based trust management has been proposed in the literature as a mechanism to filter out the misbehaving nodes while searching for a packet delivery route. However, building a trust model that adopts recommendations by other nodes in the network is a challenging problem due to the risk of dishonest recommendations like bad-mouthing, ballot-stuffing, and collusion. This paper investigates the problems related to attacks posed by misbehaving nodes while propagating recommendations in the existing trust models. We propose a recommendation based trust model with a defence scheme, which utilises clustering technique to dynamically filter out attacks related to dishonest recommendations between certain time based on number of interactions, compatibility of information and closeness between the nodes. The model is empirically tested under several mobile and disconnected topologies in which nodes experience changes in their neighbourhood leading to frequent route changes. The empirical analysis demonstrates robustness and accuracy of the trust model in a dynamic MANET environment.

Index Terms—Dishonest recommendation, filtering algorithm, mobile ad hoc networks, recommendation attacks, recommendation management, Trust management models

1 Introduction

Mobile ad hoc networks (MANETs) are characterised by the lack of infrastructure (i.e. pre-existing communication backbone) and central authority (such as base stations or mobile switching centres) to establish and facilitate communication in the network [1]. It is composed of a set of autonomous devices that work as network nodes agreeing to relay packets for each other and have dynamic topologies, with resource constraints, and limited physical security [2]. MANETs' applications are increasing in future network paradigms including vehicular and mesh networks. Many civilian and military services are demanding MANET applications, ranging from emergency rescue services such as hurricane and earthquake disasters to exchanging critical information on the battlefield or even home and personal area networking [3]. The formation and sustained existence of MANET services are mainly based on an individual node's cooperation in packet forwarding. Due to the unique characteristics and demanding use, MANETs are vulnerable to attacks launched by misbehaving nodes [2]. One of the approved mechanisms to improve security in MANETs is to use trust management techniques to deal with the misbehaving nodes and stimulate them to cooperate [4].

Trust as a social concept can be defined as the degree of subjective belief about the behaviour of a particular entity [5]. Trust is being increasingly adopted as an important concept to design and analyse security problems in distributed systems to guide decision making [6]. Trust in MANETs is the opinion held by one node (known as evaluating node) about another node (known as evaluated node), based upon the node's past behaviour and recommendations from other nodes (known as recommending nodes) in the network.

Existing trust management frameworks for MANETs can be categorised into two types. The first establishes trust relationships between nodes based on direct interactions only [7], [8]. The second type is based on direct observations of the node itself and recommendations provided by other nodes in the network [9], [10]. The use of recommendation based trust technique can be advantageous to nodes in discovering misbehaving nodes prior to interaction, thus avoiding a potential bad experience. Using recommendations, nodes in MANETs can make more informed decision on the selection of routing path even if they did not have any direct interactions in the past [9]. Being acquainted with several distant nodes (not neighbours) can be done sending a single packet to them, and it could help in saving energy [11].

Together with the advantages comes the challenge of handling dishonest recommendations in MANETs. In absence of past interactions, a particular node might not be well informed to make an assessment of trustworthiness of another node. In such cases, the evaluating node solicits recommendations from the evaluated node's neighbours (acquaintances) with whom it has a history of interaction. However, to maximise the gain of individual and their

- *A.M. Shabut and I.U. Awan are with the School of Engineering and Informatics, University of Bradford, Bradford, United Kingdom. E-mail: A.R.M.Shabut@student.bradford.ac.uk, I.U.Awan@bradford.ac.uk.*
- *K.P. Dahal is with the School of Computing, University of the West of Scotland, Glasgow, United Kingdom. E-mail: keshav.dahal@uws.ac.uk.*
- *S.K. Bista is with the CSIRO Computational Informatics, Canberra, Australia. E-mail: sanat.bista@csiro.au.*

Manuscript received 30 Nov. 2013; revised 5 Nov. 2014; accepted 16 Nov. 2014. Date of publication 8 Dec. 2014; date of current version 31 Aug. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below. Digital Object Identifier no. 10.1109/TMC.2014.2374154

acquaintances, nodes could resort to dishonest behaviours through attacks such as ballot stuffing, bad-mouthing or colluding (refer Section 3 for details on attacks). Such attacks could eventually lead to trust framework malfunction [12].

Solutions proposed to tackle these problems are limited and not adequately effective [11], [12], [13], [14], [15], [16]. For instance, one of the approaches [11] judges the honesty of the recommending node by referring to their trust values. A recommending node with a high trust value is preferred and seen as a trustworthy one. However, a node can be trustworthy in terms of packet forwarding but could be a bad node as a recommending node. Filtering out dishonest recommending nodes becomes a serious problem when recommending nodes collude with each other to accomplish a malicious goal [12]. This may result in confusing and misleading trust model in judging the nodes' trustworthiness.

To overcome some of these limitations, this paper proposes a recommendation based trust model with a defence scheme to filter out attacks related to dishonest recommendations like bad-mouthing, ballot-stuffing, and collusion for mobile ad hoc networks. The recommending node is chosen based on three factors to check its honesty: number of interactions with the evaluated node, unity of view with the evaluating node for solving the problem of the scarcity of knowledge, closeness to the evaluating node. Recommendations are accumulated over a period of time to ensure the consistency of recommendations provided by a recommending node regarding the evaluated node. Clustering technique is adopted to dynamically filter out recommendations between certain timeframe based on: a). Number of interactions (using confidence value), b). Compatibility of information with the evaluated node (through deviation test) and c). Closeness between the nodes. Different nodes are chosen in the evaluation procedure to test the performance of the filtering algorithm against various mobile topologies and neighbourhoods.

2 Related Work

In recent years, different trust and reputation models have been proposed to enhance security in MANETs to enable nodes to evaluate their neighbours directly or through recommendations from other nodes in the network. Though the proposed models have paid some attention to the problem of dishonest recommendations, finding out effective mechanisms to eliminate or mitigate the influence is still a challenging problem for MANETs.

CONFIDANT [17] uses the personal experience mechanism to deal with the problem of dishonest recommendation. It applies the deviation test on the received recommendations and excludes the ones deviating above the threshold value. The reputation value of a recommending node is updated based on the results from the deviation test. The model cannot prevent the dissemination of false recommendation and negative recommendation is the only information exchanged between nodes [18]. Michiardi and Molva [19] propose CORE model, which only accepts positive recommendation by others. Consequently, this can lead to decreased efficiency of the system because nodes cannot exchange bad experiences from the misbehaving ones in the network. Also, CORE cannot be resilient against ballot-stuffing attack as it leaves ways for misbehaving nodes to collude and gain unfair high ratings. Wang et al. [20] propose a trust-based incentive model for self-policing mobile ad hoc networks to reduce the impact of false recommendation on the accuracy of trust value. However, the performance of the model is not tested against specific attacks such as bad-mouthing. Authors in [21] propose RFSTrust, a trust model based on fuzzy recommendation similarity, which is presented to quantify and evaluate the trustworthiness of nodes. They use similarity theory to evaluate the recommendation relationships between nodes. That is, the higher the degree of similarity between the evaluating node and the recommending node, the more consistent is the evaluation between the two nodes. In this model, only one type of situation is considered when selfish nodes attack is present and the performance of the model is not tested against other attacks related to recommendation. Soltanali et al. in [22], propose a model of trust to encourage the cooperation between nodes by using direct observation and recommendation. This model only accepts the last opinion of a node, which is passed to a reputation manager system at the end of each interval. Considering only the last opinion is not insightful enough to recognise the fluctuation in node's behaviour, like in on-off attack [12]. In an attempt to increase the honesty of utilising recommendations, Li et al. in [10] include a confidence value in their evaluation by combining two values: trust and confidence into a single value called trustworthiness. They utilise the trustworthiness value to put weight on recommendations in which a recommending node with higher trustworthiness value is given more weight. Collusion attack in providing false recommendation is not considered by this work, and this may cause incorrect evaluation of the received recommendations [5]. Hermes [13] is a recommendation based trust model that uses an additional parameter known as an acceptability threshold (in relation to the confidence level). The notion of acceptability is used in the computation of recommendation to ensure that adequate observations of the behaviour of participating node has been obtained. However, the selection of acceptability is a trade-off between obtaining more accurate trustworthiness value and the convergence time required to obtain it. A recommendation exchange protocol (REP) is proposed by Velloso et al. [23] to allow nodes to send and receive recommendations from neighbouring nodes. It introduces the concept of relationship maturity based on how long nodes have known each other. Recommendations forwarded by long term associates are weighed higher than that from short term associates. The maturity of relationship is evaluated on the basis of a single factor by considering only the duration of relationship. Yu et al. in [24] propose a clustering technique to filter out trustworthy recommendations from untrustworthy ones. They follow the majority rule by selecting the cluster with the largest number of recommendations as trustworthy one. They tested their model against some attacks like bad mouthing and ballot stuffing. However, majority rule could actually be harmful as some nodes can collude to perform an attack, and not provide an honest judgment about other nodes.

The aforementioned discussion highlights limitations of the trust models in their abilities to shield nodes from malicious behaviour in the network. It can be seen from the

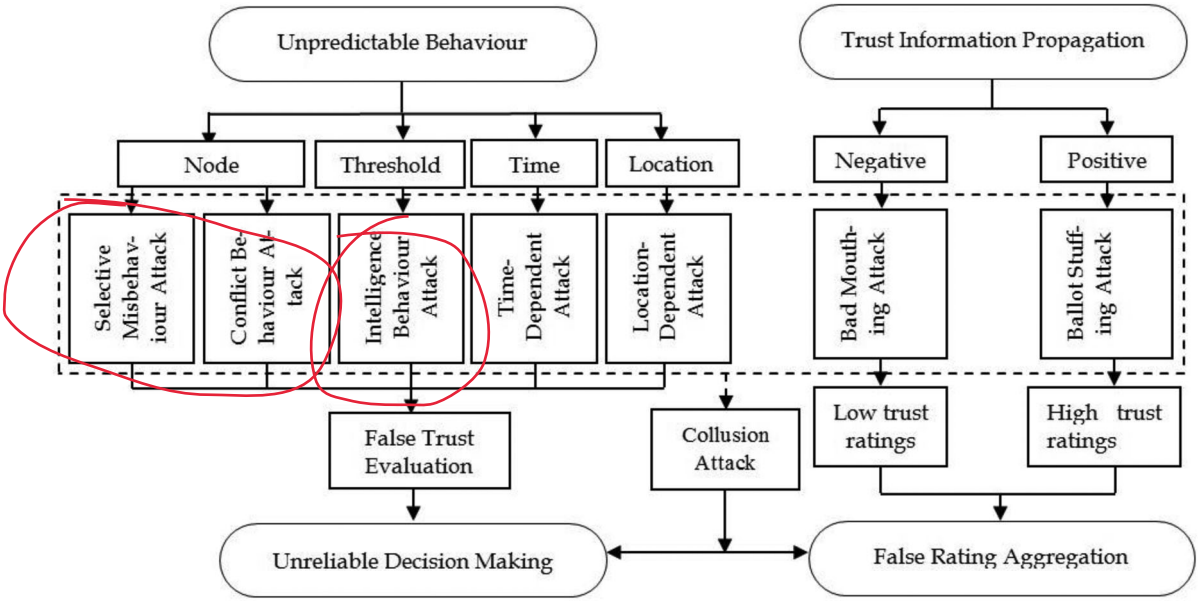


Fig. 1. Attacks related to misbehaviour problems in recommendation management within trust and reputation frameworks.

literature review that most of the models relied on single parameter to compute trustworthiness. To address these limitations, a defence scheme is proposed in this paper using multiple parameters (as specified in Section 1) to compute the trustworthiness of recommenders. The model underlines the importance of social properties in evaluating trustworthiness and uses it in investigating the relation between closeness of nodes and similarity in behaviour. The use of proof of time and location, missing in the current literature, is considered by the proposed model. False negative and false positive problems in evaluating the recommendation's trustworthiness and their impact on the network performance are thoroughly investigated.

评价可信任程度trustworthiness：强调social属性, clossness-similarity, // time和loc位置的证据

3 Attacks Related to Recommendation Management in Trust and Reputation Frameworks

It is indeed a challenge to safe guard a network against wide range of attacks. Recent focus of research in this area has been on the problems associated with misbehaving nodes in the context of packet forwarding, like blackhole or wormhole attack [25]. For quality assurance, it is important that trust management frameworks be resilient to attacks [10]. Although several research have put considerable effort to protect the propagation and aggregation of recommendations in a trust model, research is still in its early stages [12]. The following attacks, namely, bad mouthing attack, ballot stuffing attack (BSA), selective misbehaviour attack, intelligent behaviour attack, time-dependent attack (TDA) and location-dependent attack (LDA) (see Fig. 1 for the classification of attacks), are targeted at the propagation and aggregation of recommendation [10], [12], [26]. Location-dependent attack is used for the first time in this paper. The attack behaviours are summarised below:

- *Bad mouthing attack (BMA).* In this type of attack, conspiring nodes propagate unfairly negative ratings of good nodes with an ill intent to tarnish their reputation in the network. Such collusive behaviour may lead to the blocking of valid paths in the network by confusing the trust and reputation management mechanism.
- *Ballot stuffing attack (BSA).* Propagation of unfairly positive ratings for some poorly performing nodes by collusive nodes in the network lead to ballot stuffing attack. The intention of collusive nodes is to mislead the trust mechanism and cause it to malfunction in accurately reporting the trustworthiness of assessed node. 只向部分节点提供false rating假评价
- *Selective misbehaviour attack (SMA).* This attack victimises some trusted nodes by propagating false ratings for them, while at the same time acting normal to other nodes. This type of behaviour can be very difficult to detect for the trust mech 根据阈值选择threshold
- *Intelligent behaviour attack (IBA).* This attack selectively provides recommendation with high or low ratings according to the trust threshold. This kind of attack can cause malfunction to the trust framework by dynamically responding to trust threshold and behaving based on it. 有时behave正常 又是misbehave
- *Time-dependent attack.* This attack makes participating nodes to change their behaviour by time. Nodes can behave normally for a period of time and can misbehave by providing unfair ratings at other times. This attack also has its roots in the subjective property of trust.
- *Location-dependent attack.* This attack exploits mobility property of MANETs, where a node behaves differently according to its location. This attack originates from the subjective property of trust where behaviours at one location cannot affect evaluating trustworthiness of nodes at another location.

The summarised attacks belong to two categories: false rating (BMA, BSA, and SMA), and inconsistent rating based on the trust threshold, time, or location (IBA, TDA, and LDA). Some of the countermeasures illustrated below can be used for both categories or being specifically designed for one category. For example, [19] proposes the use of only

positive recommendations, while [17] uses only negative recommendations and this can countermeasure attacks like ballot stuffing and bad mouthing. This kind of defence can be harmful to trust information because nodes cannot report their complete experiences. Statistical methods like Bayesian theory to accurately compute the correctness of recommendations can be a proper solution to both categories [26]. Proof of sufficient interactions [13], and specifying a certain threshold of negative and positive recommendation, besides, the majority opinion technique [24] could also be used to mitigate the effect of false and inconsistent rating. Comparison between recommendation list and proof of time and location of the recommendation provider is also a promising solution to time and location-dependent attacks. The method of comparing time and location is considered first time in the proposed algorithm.

What follows from above discussion is that the recommending nodes' trustworthiness cannot be assessed by just a single scheme. It should be supported by using many behavioural and social properties (such as, the closeness between nodes, and proof of time and location), which is missing in the existing literature. In order to improve accuracy and robustness of the trust model, the influence of the untrustworthy recommendations should be mitigated to overcome the problem of false negative and false positive.

4 The Proposed Model

We propose a recommendation-based trust management model to secure the routing protocol between source and destination nodes based on the trust value of each node in the path. The model considers the problem of the attacks discussed earlier due to some misbehaving nodes in MANETs. We make use of a Bayesian statistical approach similar to that used in [27] for computing trust values based on the assumption that they follow a beta probability distribution. Beta distribution is estimated by using two parameters (α, β). They can be calculated by accumulating observations of forwarding and dropping packets where α represents the accumulation of positive observations (forwarded packets) and β represents the accumulation of negative observations (dropped packets). The beta distribution can be defined by gamma function as shown in Eq. (1):

$$f(p|\alpha,\beta) = \frac{\Gamma(\alpha+\beta)}{\Gamma(\alpha)(\beta)} p^{\alpha-1}(1-p)^{\beta-1}, \quad (1)$$

where $0 \leq p \leq 1; \alpha, \beta > 0$ with a condition that $p \neq 0$ if $\alpha < 1$ *and* $p \neq 1$ *if* $\beta < 1$.

Nodes in the network observe each other's behaviour in order to construct a trust relationship representing the degree of trustworthiness one node can put on another. These relationships are useful to help nodes decide whether to forward packets to a specific neighbour or not. In the proposed model, an initial trust relationship is established between two nodes i and j as $(\alpha_{ij}, \beta_{ij})$ at time t, where α_{ij} denotes the positive interactions observed by node i about node j, and β_{ij} denotes the negative interactions observed by node i about j.

The trust model computation needs evaluating nodes to rate other evaluated nodes to find trustworthy neighbour to assign network activities. If nodes has no initial value to put on another node, trust predictions can not be made and this lead to the appearance of cold-start. Thus, an initial constant value of trust can help side stepped the problem. To overcome the cold-start problem (which arises when nodes have no historical trust profile i.e., no interactions such as rating), at time $t = 0$, we start with $\alpha_{ij} = \beta_{ij} = 1$, which assigns a value of 0.5 to the initial trust held by node i about node j. This value can be translated as complete uncertainty about the distribution of the parameter which means no observation or evidence has been collected. If the estimated positive and negative interactions between two nodes i and j are denoted as ρ and n respectively, α_{ij} and β_{ij} would be calculated as $\alpha_{ij} = \rho + 1$ and $\beta_{ij} = n + 1$ where ρ and $n \geq 0$. After each observation, the trust metric can be computed and updated from these parameters as the expectation of beta distribution given by $\frac{\alpha_{ij}}{\alpha_{ij}+\beta_{ij}}$.

With the increase of mobile nodes and resources in MANETs, the difference in rating scale between different nodes becomes an issue which has led to a data sparsity problem. Data sparsity in a recommendation-based trust model is occurred in a situation of lacking or insufficient interaction experience in the early time of establishing the network, or when most of the nodes are inactive in recommendation. It is considered as one of the main challenges for the high quality of recommendation in trust research field for MANETs. Several solutions have been proposed to overcome the problem of data sparsity in MANETs. This can be categorised as: a) methods that utilise similarity metrics to enhance the selection of recommendations from similar neighbours [21], [28], b) methods that implement aggregation techniques to integrate the ratings given by all the neighbours [28], and c) methods using data imputation to improve the selection of missing or insufficient ratings of neighbours [29]. To overcome this problem, the proposed trust model uses the nearest neighbour clustering technique [24], [29] to impute ratings and reduce the sparsity. Besides, it can improve the consistency of received recommendations of the filtering algorithm. For example, recommendations from a misbehaving node can have a range of multiple different ratings for the evaluated node. These ratings may be inconsistent in which they can differ from each other in a short period of time, a malicious act of the misbehaving node to confuse the trust model.

Dynamic clustering of the recommendations over a period of time can filter out deviated ratings from the list of recommendations, thus decreasing the influence of false estimations in computing trust value. This is achieved by maximising the information contained in the neighbourhood of nodes and minimising the errors in imputation of rating values. The proposed model clusters recommendations based on three different criteria: (a) number of interactions by the means of using confidence value, (b) compatibility of information with the evaluated node by the means of deviation test, and (c) closeness between these nodes. The use of multiple criteria to judge whether a node is dishonest can mitigate the influence of false negative and false positive ratings. Furthermore, the neighbourhood relationships between nodes are better predicted and identified using the proposed multiple criteria. For example, confidence is used to solve the

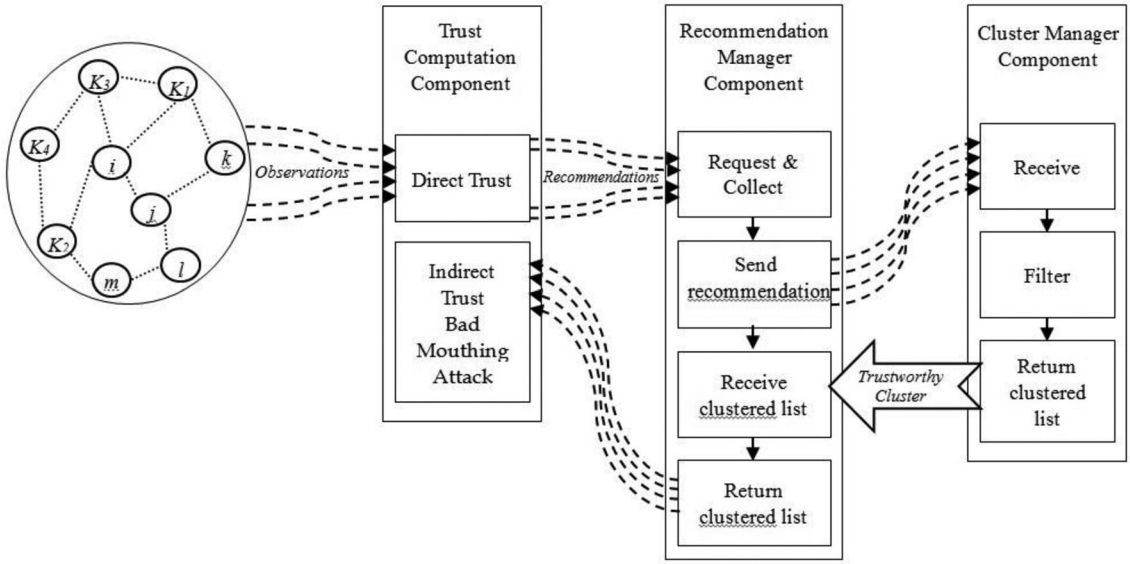


Fig. 2. Recommendation-based trust model components.

problem of missing and insufficient ratings in recommendation list, deviation is to unify the received ratings, and closeness is to ensure similarity in preferences and environmental conditions of nodes. As a result, the proposed filtering algorithm is able to effectively solve the problem of data sparsity at less cost than massive similarity calculation for the ratings in the received recommendations.

The model has three components deployed to evaluate trust: (a) *Trust Computation Component* that uses direct as well as indirect (second hand) trust information. (b) *Recommendation Manager Component* that requests and gathers recommendations for a node from a list of recommending nodes, and (c) C*luster Manager Component* which filters out dishonest recommendations from the list and sends out a list of trustworthy recommendations to the manager component. Fig. 2 shows the model's components and their interaction process. The recommendation manager and cluster manager components are described in Section 5.

The *trust computation component* obtains direct trust value from two nodes that have already initiated a trust relationship. These two nodes can continue to interact with each other at least for a period of time they are within the range. Direct trust value is considered to be accurate and its computation invulnerable to dishonest recommendations. Direct trust value T_{ij}^{d} of node i about j is calculated as in Eq. (2):

$$T_{ij}^{d} = \frac{\alpha_{ij}}{\alpha_{ij} + \beta_{ij}}. \tag{2}$$

衰减因子 decay factor 以减少过去经验的影响

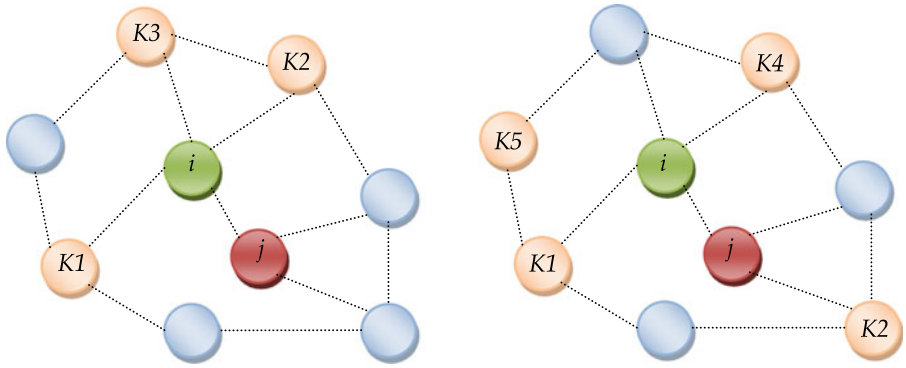
Influence of past experiences change over time in a dynamic environment. It is thus important for a trust model to consider this change in influence. The proposed model incorporates a decay factor (μ) to gradually decrease the influence of past experience over time, prior to the aggregation with new trust values. Forgetting of past experiences is carried out by adjusting the time frame of observations while recording the positive or negative experience. However, trust decays over time even during inactivity periods and it is thus important to consider the diminishing impact of trust over the time. The first situation is when a node observes an additional new positive or negative interaction between time t_i and t_{i+1} denoted as ρ^{new} and n^{new}, then the updated ρ and n should be reduced by the decay factor μ before merging them with the new values. Therefore, at time t_{i+1}, ρ and n is updated respectively according to the formula in Eq. (3):

$$\rho = \rho^{old} * \mu + \rho^{new}, \quad n = n^{old} * \mu + n^{new}, \tag{3}$$

where $0 \leq \mu \leq 1$, ρ^{old} and n^{old} are the old positive and negative experiences observed by the node. The second situation is when there is no observed new positive and negative interaction between time t_i and t_{i+1}, then, at time t_{i+1}, ρ and n is updated respectively as in Eq. (4):

$$\rho = \rho^{old} * \mu, \; n = n^{old} * \mu. \tag{4}$$

Indirect trust needs to be considered, when two nodes have not established a previous trust relationship through exchange of packets or any other form of communication. In such case, the evaluating node suffers from the sparsity problem in which it lacks interaction information to judge the trustworthiness of the other node being evaluated. Indirect trust is also calculated using the beta-function, similarly like the direct trust was computed earlier. Indirect trust is actually the direct observations obtained by one node about its neighbours which can be used by another node as second-hand information. The utilisation of indirect trust to predict other nodes' trustworthiness can help overcome the limitations of filtering algorithms regarding data sparsity and cold-start problems when direct trust is not existed. Indirect trust propagation can improve coverage by allowing new nodes to perceive several recommendations to predict the trustworthiness of nodes which have not interacted before. We can say that node k's direct observations of node j could be indirect or second hand information to another node i (given that node i and j have not interacted in the past). Therefore, indirect trust value is calculated using $(\alpha'_{ij},\beta'_{ij})$ and updated by two variables: ρ', describing the



(a) Recommending nodes at time t_i (b) Recommending nodes at time t_{i+1}

Fig. 3. Recommendation by time.

number of positive interactions, and n', describing the number of negative interactions. Further, α'_{ij} and β'_{ij} are calculated as $\alpha'_{ij} = \alpha'_{ij} + \rho'$ and $\beta'_{ij} = \beta'_{ij} + n$. If the evaluating node i receives N recommendations for the evaluated node j denoted by $k = 1, 2, \ldots, N$, indirect trust T^i_{ij} of node i about j is calculated according to the Eq. (5):

$$T^i_{ij} = \sum_{k=1}^{N} \frac{\alpha'_{kj}}{\alpha'_{kj} + \beta'_{kj}}. \quad (5)$$

k个推荐/间接信息 直接sum在一起？

While indirect trust information is important to incorporate in a trust model for MANETs, involving this kind of information can be vulnerable to intentionally generated dishonest recommendations.

For each node in the network, trust value T_{ij} is calculated by combining both direct and indirect trust values with different weights denoted by w_d and w_i respectively. T_{ij} is computed according to Eq. (6):

$$T_{ij} = w_d * T^d_{ij} + w_i * T^i_{ij}, \quad (6)$$

where $w_d + w_i = 1$. The weights are used because of their significant impact on diminishing the possibility of wrong trustworthiness evaluation of direct and indirect trust information by nodes. In most of existing models, higher weight is usually given to the direct information as it is less prone to dishonest recommendation. However, MANETs' characteristics such as mobility and frequent change in topology make it difficult to completely trust the source of information even if it is the nodes self-assessment. The weight in this model is dynamically calculated based on the quantity and quality of interactions observed by evaluating nodes. If the evaluating node has enough experience about the evaluated node and the evaluated node is not compromised or prone to any environmental conditions (e.g. node failure, or low energy level), it is given equal or more weight than indirect information. While, if the evaluating node is not able to judge the trustworthiness of the evaluated node, more weight is given to the indirect trust.

5 Cluster-Based Recommendation Filtering

This section analyses the functionalities of recommendation and cluster manager components and shows how they

本章 过滤掉不诚实的推荐

work together to filter out untrustworthy recommendations. The proposed filtering technique takes into consideration the dynamic characteristics of MANETs that change over time. The honesty of recommending nodes is evaluated over a period of time to mitigate the influence of bad behaviour of the same node over time. Fig. 3 shows the dynamic topology of MANETs. Consider that, a node i wants to evaluate another node j by requesting recommendations from its neighbours. The evaluating node i receives a list of recommending nodes referred as $\{k_1, k_2, k_3, \ldots, k_N\}$. At time t_i (refer Fig. 3a), the location and number of recommending nodes differ from the recommending nodes at time t_{i+1} as shown in Fig. 3b.

Recommendation manager in the proposed model works as an intermediate component between indirect trust computation and cluster manager components. It helps in detecting and eliminating false recommendations. Recommendation manager has three important roles: 1) send recommendation request to the evaluating node's neighbours; (2) collect received recommendation and send it to the cluster manger which runs the filtering procedure; (3) receive the filtered recommendation and send it back to the trust computation component. Recommendation manager requests and gathers recommendation list for an evaluating node i about node j from a list of recommending nodes $\{k_1, k_2, k_3, \ldots, k_N\}$ between time t_i and t_{i+1} and send it to the cluster manager to run the filtering algorithm. After filtering, it receives the trustworthy clusters as a list of honest recommendations denoted as $\{k^{Tr}_1, k^{Tr}_2, k^{Tr}_3, \ldots, k^{Tr}_N\}$. The final task is to send the trustworthy cluster $C^{Trustworthy}$ to the requesting node. Algorithm 1 illustrates the recommendation manager algorithm.

Algorithm 1. Recommendation Manager Algorithm

1. ***For*** each recommendation request ***Do***
2. ***Send*** request to neighbours
3. ***Collect*** received recommendations
4. ***Construct*** $L = \{k_1, k_2, k_3, \ldots, k_N\}$
5. ***Send*** L to the cluster manager for processing
6. ***Receive*** trustworthy cluster $C^{Trustworthy} = \{k^{Tr}_1, k^{Tr}_2, k^{Tr}_3, \ldots, k^{Tr}_N\}$
7. ***Send*** $C^{Trustworthy}$ to the requesting node
8. ***End*** For

TABLE 1
Levels of Confidence for the Proposed Model and TMUC Model with the Same Trust Levels

α	B	s	f	Trust value	Confidence value (proposed model)	Confidence value (TMUC model)
1	1	0	0	0.5	0	0.916666667
5	2	4	1	0.714285714	0.446716665	0.974489796
10	4	9	3	0.714285714	0.595938982	0.986394558
15	6	14	5	0.714285714	0.666357595	0.990723562
20	8	19	7	0.714285714	0.709401356	0.992962702
25	10	24	9	0.714285714	0.739179735	0.994331066
30	12	29	11	0.714285714	0.761351694	0.995253916
35	14	34	13	0.714285714	0.778686666	0.995918367
40	16	39	15	0.714285714	0.792721071	0.996419620
45	18	44	17	0.714285714	0.804384801	0.996811224
50	20	49	19	0.714285714	0.814277976	0.997125611

Nodes are clustered based on three values, namely: *confidence value*, *deviation value*, and *closeness value*. The following sections will explain these values and give an overview of the clustering process and its algorithm.

5.1 Confidence Value V_{ik}^{conf}

The notion of confidence was introduced in [30] where confidence value and trust value are combined together to derive a single trustworthiness value of a node. Following that, trust models in [10], [13], [31] have also considered the confidence value as a desired parameter to achieve a single trust value to represent the trustworthiness of nodes. Confidence value can be used to solve the problem of short-term and long-term observations. That is, nodes may have the same level of trust with different number of observations. For example, the trust value of a node at the initial time with $\alpha = \beta = 1$ is 0.5, and after a sequence of positive and negative interactions in which $\alpha = \beta = 50$, the node has the same trust value of 0.5 about the evaluated node (see Table 1 for more information). Confidence value starts from 0 in case of no observations between nodes and increases gradually with the number of recorded observations. Relying only on the trust value can raise the problem of short-term and long-term observations. Nodes in the network can have nearly the same level of trust though they may have different levels of observations. Consequently, this can lead to wrong estimation in judging the ability of nodes to be honest recommending node.

The proposed filtering algorithm clusters recommending nodes based on the level of confidence for two reasons. Firstly, the nodes with higher confidence value (those having sufficient interactions with evaluated node) are desirable because the higher number of interactions will offer rich information that would help in choosing better recommending nodes. Secondly, the recommending nodes with very high confidence value in the early rounds in the network (when there are no enough interactions) are more likely to be attackers. Consequently, it may lead to exclusion of dishonest nodes from the recommendations list in early stages. The confidence value is computed as the variance of the beta distribution with some modifications as in [10] and [13]. Nodes use the confidence value to make a correct decision about the trustworthiness of recommending nodes taking into account the number of observations accumulated by each node. Suppose that i is an evaluating node that received recommendations from a recommending node k, confidence value V_{ik}^{conf} is calculated as in Eq. (7):

累积数量太少，则不可信

$$V_{ik}^{conf} = 1 - \sqrt{12}\sigma_{ik},$$
$$V_{ik}^{conf} = 1 - \sqrt{\frac{12\alpha_{ik}\beta_{ik}}{(\alpha_{ik}+\beta_{ik})^2(\alpha_{ik}+\beta_{ik}+1)}}, \quad (7)$$

where σ_{ik} is the beta distribuation varience between i and k, α_{ik} and β_{ik} is the accumulated positive and negative interactions between i and k.

Using this formula the value of confidence falls between the interval of [0, 1], where 0 means that no previous interactions are recorded between the evaluating and evaluated node while 1 means complete confidence in the evaluated node. The rational of using and computing the confidence value is shown in Fig. 4. We compare the confidence value computed using the proposed method with that in [32] (we call it TMUC for short), which computes the confidence value using only the standard deviation. The proposed computation method of confidence value can effectively reflect the knowledge held by nodes based on the number of interactions better than the calculation in TMUC. For example, when $\alpha = \beta = 1$ which means there is no previous interaction between two nodes, the proposed method of computing confidence value is 0 while in TMUC, it is nearly 0.91 which is a high value close to 1. Starting with high confidence value in case of no interactions can confuse the trust mechanism and prevent it from making good judgement

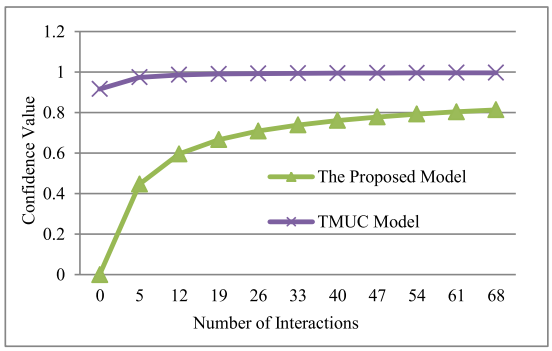


Fig. 4 Relationships between Interactions and Confidence for the proposed model and TMUC model.

about behaviour of the evaluated node. Table 1 shows the values of positive and negative interactions and the confidence value for each level of interaction for both the proposed model and the work in TMUC. Fig. 4 explains the relationship between interactions and the level of confidence when the trust levels are the same.

From Fig. 4, it can be seen that the proposed method of computing confidence offers a better range for the confidence value as compared to that by TMUC. This variation reflects better accumulated interactions when the trust values (refer Table 1) are same. When there are no interactions, confidence value from the proposed model is 0 and it progresses with the increasing number of interactions. Whereas with TMUC, the confidence value is already at 0.91 in case of no interactions and thus is nearly at saturation level when number of interactions more than 19.

评价的偏差 太大或太小

超过阈值就被过滤

5.2 Deviation Value V_{ij}^{dev}

Deviation value represents to what extent the received recommendation is compatible with the personal experience of evaluating node. This value has been used by the means of the deviation test in [17] to ensure the unity of view with the receiving node. Each node compares received recommendation with its own first-hand information and accepts only those not deviating too much from self-observations. In the proposed model the deviation value is used as an additional parameter in the clustering algorithm to filter out any recommendations deviating beyond a predefined deviation threshold. A problem that could arise here is when the evaluating node lacks historical information for interactions with the evaluated node, thus not providing a base value for comparison. In order to overcome this problem, the proposed method compares the confidence level of the evaluating node with that of the recommending node. The confidence value is calculated using Eq. (7). The deviation test is only applied if both nodes have similar level of confidence. Assume that there are three nodes (i, j and k), and node i attempts to calculate the trust value of its neighbour node j using recommendation provided by node k. In this scenario, node i first compares its confidence level which denoted as $Conf\_Level$ with the recommending node as in Eq. (8). If the confidence difference is less than a threshold value denoted as $Conf_{Threshold}$, then node i calculates the deviation value as a difference between the receiving recommendation and direct observations of the evaluated node as held by the evaluating node as in Eq. (9). The resulting value is compared to a predefined deviation threshold d and we exclude any recommendations that differ widely from the evaluating node's own information

confidence的偏差 recommendation的偏差 排查偏差太大的

$$Conf\_Level = |CV_{ij} - CV_{kj}| \leq Conf\_Threshold, \quad (8)$$

where CV_{ij} is the confidence value of i about j, and CV_{kj} is the confidence value of k about j. If the Eq. (8) is successful, deviation value V_{ij}^{dev} is calculated as follows:

$$V_{ij}^{dev} = \left|T_{ij}^{d} - T_{kj}^{r}\right| \leq d^{dev}, \quad (9)$$

where T_{ij}^{d} is the direct trust value of i about j, and T_{kj}^{r} is the received trust value of k about j.

5.3 Closeness Centrality Value V_{ij}^{close}

空间位置的紧密

Trust is a social concept and it is thus possible to apply the perceptions of social life in trust computation and recommendation propagation. An interesting direction of trust research in MANETs is to utilise social relationships in evaluating trust among nodes in a group setting by employing the concept of social structures [5]. The proposed model uses the concept of closeness centrality between the evaluating nodes and the recommending node from the social trust. Closeness centrality [33] measures the distance between the evaluated node and the recommending node in terms of physical distance, number of hops, or delays. In the proposed model closeness centrality is a measure of the distance between the evaluating node and the recommending node. The use of the closeness centrality enhances the filtering algorithm as close nodes are likely to possess same nature and counter nearly same environmental and operational conditions over a period of time in the network. Furthermore, close friends may have more interactions in the time of friendship. Consequently, trust values for the close neighbours converge to nearly same level. This may help in recognising the untrustworthy recommending node whose recommendation is much different from the close recommending nodes. Closeness value V_{ij}^{close} refers to the degree of node i's closeness to a recommending node k at time t and is calculated by Eq. (10):

$$V_{ik}^{close} = \sqrt{\left(x_i^{loc} - x_k^{loc}\right)^2 + \left(y_i^{loc} - y_k^{loc}\right)^2} \leq d^{dis}, \quad (10)$$

where (x_i^{loc}, y_i^{loc}), (x_k^{loc}, y_k^{loc}) are the positions of node i and node k at time t and d^{dis} is a predefined distance threshold between node i and node k which should be less than the transmission range.

紧密的节点更可能相同的属性和计数

5.4 Cluster Procedure

The cluster manager in the proposed model receives a list of recommendations from the recommendation manager and processes it using a clustering technique. The clustering algorithm is run by the evaluating node on all the recommendations in the list $L = \{k_1, k_2, k_3, \ldots, k_n\}$. A vector of three values $(V_{ij}^{conf}, V_{ij}^{dev}, V_{ij}^{close})$ is provided by a recommending node for the clustering operation. The clustering algorithm divides the vectors from the recommending nodes into a predefined number of clusters denoted as K. Initially each vector is considered as a cluster, and then two clusters with the shortest Euclidean distance are merged together to produce a new cluster. The clustering process is repeated by merging two clusters from the previous iteration until the predefined number of clusters K is reached. The first step of the clustering process aims to merge vectors with the closest similarity. In the second step, it selects the trustworthy clusters if all the recommending nodes in a specified cluster satisfy the following rules:

选出其中的推荐群

$$C^{Trustworthy} = \begin{cases} R_{ij}^{Trustwothy} & if\ \left(V_{ij}^{conf} \geq d_{min}^{conf}\right) \text{and} \left(V_{ij}^{conf} \leq d_{max}^{conf}\right) \\ & if\ \left(V_{ij}^{dev} \leq d^{dev}\right) \text{and} \left(V_{ij}^{close} \leq d^{dis}\right) \\ R_{ij}^{Untrustworthy} & other\ wise, \end{cases}$$

where $R_{ij}^{Trustwothy}$ is the trustworthy recommendation, $R_{ij}^{Untrustworthy}$ is the untrustworthy recommendation, d_{min}^{conf} is

the minimum confidence threshold, d_{max}^{conf} is the maximum confidence threshold.

The next step is to apply majority rule to select the cluster with largest number of members. In the final step, trustworthy clusters are returned to the recommendation manager and to the evaluating node to update its indirect trust of the evaluated node. The proposed cluster process works as shown in Algorithm 2.

Algorithm 2. Cluster Manager Algorithm

1. ***For*** each recommendation list L ***Do***
2. ***For*** each rating vector in the list (α^r, β^r) ***Do***
3. ***Calculate*** confidence value V_{ij}^{conf} as in Eq. (7)
4. ***Calculate*** deviation value V_{ij}^{dev} as in Eqs. (8), (9)
5. ***Calculate*** closeness value V_{ij}^{close} as in Eq. (10)
6. ***Construct*** data vector as $(V_{ij}^{conf}, V_{ij}^{dev}, V_{ij}^{close})$
7. ***End*** For
8. ***Initialize*** each vector as a unique cluster
9. ***Repeat***
10. ***For*** each vector ***Do***
11. ***Merge*** two clusters with the shortest Euclidean distance
12. ***End*** For
13. ***Until*** number of clusters $= K$
14. ***For*** each cluster that appeared in the previous iteration ***Do***
15. ***If*** $(V_{ij}^{conf} \geq d_{min}^{conf})$ and $(V_{ij}^{conf} \leq d_{max}^{conf})$ ***Then***
16. ***If*** $(V_{ij}^{dev} \leq d^{dev})$ and $(V_{ij}^{close} \leq d^{dis})$ ***Then***
17. ***Select*** trustworthy cluster
18. ***End*** If
19. ***End*** If
20. ***End*** For
21. ***For*** each chosen trustworthy cluster ***Do***
22. ***Apply*** the majority rule
23. ***Return*** trustworthy cluster $C^{Trustworthy}$
24. ***End*** For
25. ***End*** For

6 Simulation and Results

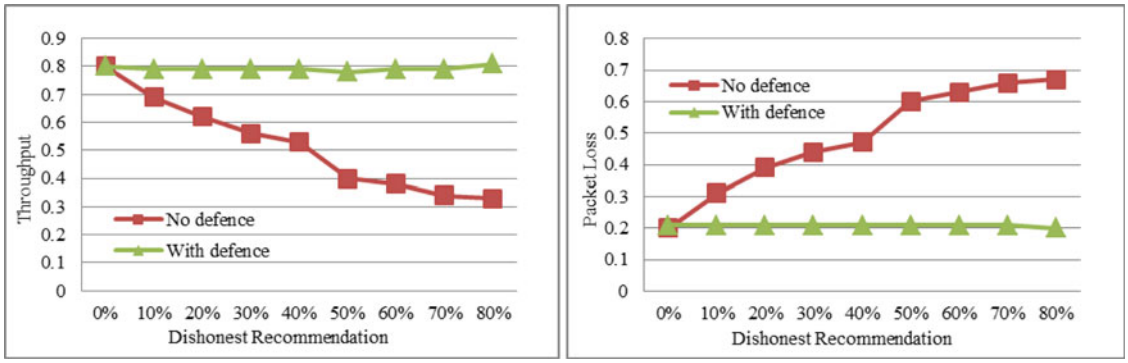
The simulation is conducted using NS2 simulator [34], an open-source discrete event simulator designed to support research in computer networking. It involves various modules to help test several network components such as packet, node, routing, application and transport layer protocol. NS2 features permit us to extend the DSR routing protocol that supports MANETs architecture. The proposed trust model components are added to the simulator to test the validity of the proposed model. A network with 50 mobile nodes is simulated randomly moving in an area of 700 × 700 square metre. Several nodes are randomly selected to provide false rating information in the form of bad-mouthing and ballot-stuffing attacks. There are 15 source-destination pairs and each source transmits 2 packets per second with a constant bit rate (CBR), and pause time 60 s, which is the time nodes need to pause to begin travelling to the next destination with a speed of 10 m/s, the packet size is 512 bytes and the simulation time is 500 s. The mobility model utilised in this paper is the random way point which is the most commonly used model in ad hoc networking research [35]. It is easy to use and movement could be considered as realistic which is very similar to the real world movement [36]. However, the proposed model can fit any other type of mobility models like RPGM model [37]. The maximum bad-mouthing and ballot-stuffing attacks percentage used in the simulation scenario is 80 percent misbehaving nodes, which is enough percentage to test these attacks. An optimistic scheme is used in choosing trust threshold value at 0.2 in which all nodes are initially expected to be trusted and normally behaving [10]. Table 2 shows the parameters used in configuring the network for the experiment. Bad-mouthing and ballot-stuffing attacks with additional permission to collude in both attacks are used in order to evaluate the defence scheme against dishonest recommendation. Number of dishonest nodes range from 0 to 80 percent and the dishonest recommendations provided deviate 50 percent from the honest recommendations. Badly behaving nodes (selfish nodes) counting to 20 percent always existed in the network and were responsible for collusion and jamming. Results from the experiment are based on multiple runs and a negligible variation is noticed.

TABLE 2
Network Configuration Parameters

Parameter	Value
Nodes	50
Area	700 m × 700 m
Speed	10 m/s
Radio Range	250 m
Movement	Random waypoint model
Routing Protocol	DSR
MAC	802.11
Source-destination pairs	15
Transmitting capacity	2 Kbps
Application	CBR
Packet size	512 B
Simulation time	500 s
Trust threshold	0.4
Publication timer	30 s
Fading timerμ	10 s
Deviation threshold d^{dev}	0.5
Conf_Threshold	0.4
d_{min}^{conf}	0.5
d_{max}^{conf}	0.9
d^{dis}	200 m

6.1 Performance Evolution

The flow of the simulation is as follows. The performance of the entire network is represented by two parameters: Network throughput and Packet loss in the presence of bad-mouthing, ballot-stuffing and selfish nodes. The trust value of a good node (not misbehaving) is evaluated against bad-mouthing attack to see the influence of such attack with and without incorporating the proposed defence scheme. The trust value of a bad node (misbehaving) is also evaluated against ballot-stuffing attack to see how such attackers can distort the trust value of this node. The performance of the proposed model in terms of recognised dishonest recommendations, false negative and false



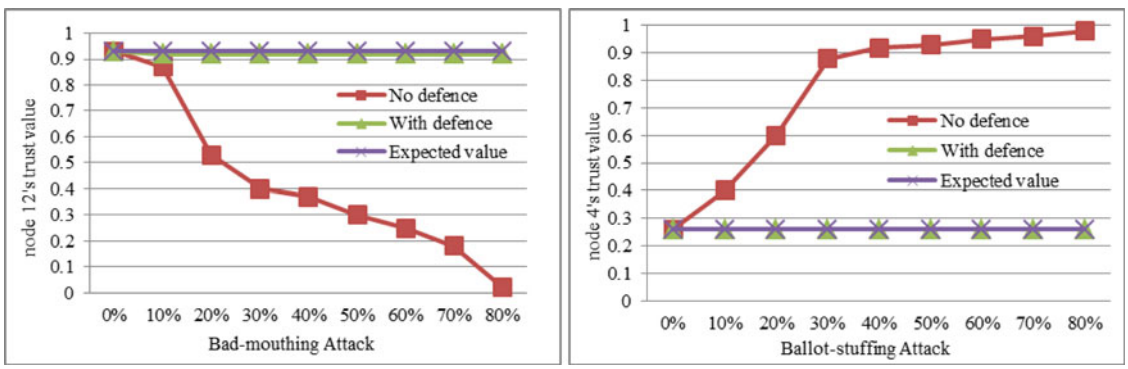
(a) Network throughput (b) Packet Loss

Fig. 5. Network performance in the presence of dishonest recommending nodes for a) network throughput; b) network packet loss.

positive in the presence of bad-mouthing attacks with and without the defence scheme is examined. Similar experiment is conducted for ballot-stuffing attack. Finally, a comparative study is conducted with the maturity model [23] proposed in the literature.

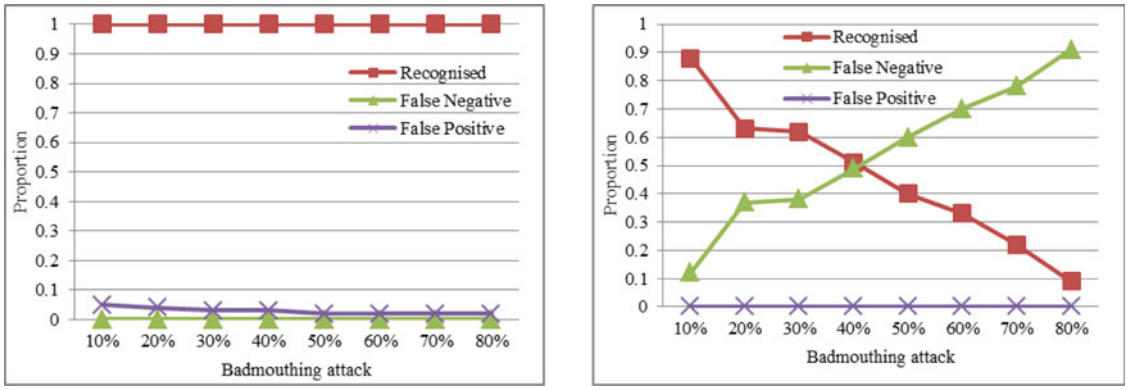
Fig. 5 demonstrates the effect of dishonest recommendation on two performance metrics; throughput and packet loss for the whole network. The *y*-axis in Fig. 5a shows the percentage of throughput, both with and without the defence scheme, in the presence of dishonest recommending nodes varying from 0 to 80 percent of the total population of nodes. It is observed that the network throughput without a defence falls from nearly 80 percent when the dishonest recommending nodes are not present to nearly 30 percent when population of the dishonest ones increases to 80 percent. Slight decrease and then increase is noticed in the throughput (Fig. 5a) for the network with defence when the percentage of dishonest recommendation nodes increases from 40 to 80 percent. This may be due to the fact that the throughput not only depends on the number of misbehaving nodes but also affected with the degree of connectivity (number of neighbours) and the ability of nodes to classify their neighbours as well as time required to achieve the classification which are different in each simulation due to network topology and mobility. However, the proposed defence mechanism was able to keep the value of throughput at nearly 80 percent even in case of higher population of the dishonest nodes. This is translated into that the defence scheme is able to mitigate the negative effect of dishonest recommendation on the throughput performance. The impact of dishonest nodes on packet loss is shown in the Fig. 5b. The percentage of packet loss rises with increasing the percentage of dishonest nodes from 20 to over 60 percent when no defence incorporated in the network. While only 20 percent packet loss can be maintained using the proposed defence scheme in the presence of dishonest recommending node that vary from 0 to 80 percent of the nodes in the network. Similarly, the percentage of packet loss decreases slightly when the percentae of dishonest recommendation nodes increases from 70 to 80 percent for the same reasons as discussed in the analysis of Fig. 5a. It can be seen from the above analysis that dishonest recommendations can significantly impact on the throughput and packet loss metrics by confusing the trust model. The proposed technique can keep those metrics at an acceptable level even when the population of dishonest nodes is high.

Fig. 6 demonstrates the average of the indirect trust held by other nodes in the network for a good node (node 12 in this case) and a bad node (node 4 in this case). The *x*-axis in Fig. 6a displays the range for the population of bad-mouthing nodes from 0 to 80 percent. The *y*-axis shows the average of the indirect trust value for a good node (node 12 in this case) as held by all the nodes that have interacted with it in the past. A comparison has been made between three different parameters as follows. First, the indirect trust value when there are no dishonest nodes, called *expected value*. Second, the indirect trust value when dishonest nodes are present and the defence scheme is working, *with defence*.



(a) Good-Node 12's Trust Value (b) Bad-Node 4's Trust Value

Fig. 6. Trust evaluation for a) good-node 12's trust value in the presence of bad-mouthing attack; b) bad-node 4's trust value in the presence of ballot-stuffing attack.



(a) With Defence (b) Without Defence

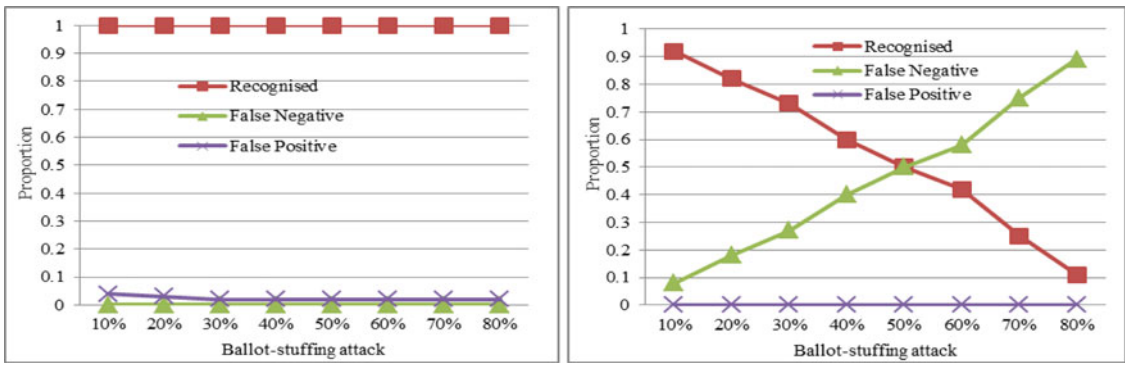
Fig. 7. Recognised, false negative and false positive proportion in the presence of bad-mouthing attack for a) with defence; b) without defence.

Third, the indirect trust value when the dishonest nodes are present and the defence technique is not working, *no defence*. It can be seen that with increasing population of badmouthing attackers, the average trust value of node 12 declines in case of *no defence*, whereas, the trust value remains the same as the *expected value* in case of *with defence*.

The effects of ballot-stuffing attack are shown in Fig. 6b. In the x-axis is the percentage of ballot-stuffing attack that varies between 0 to 80 percent and y-axis shows the values for the indirect trust compared against the same three parameter i.e. expected value, with defence and no defence cases. From the figure, it can be seen that the attacking nodes have propagated unfairly positive rating for the dishonest node (node 4) thereby raising its trust value to above 0.9 while the attacker population was 80 percent. The results here show that the defence algorithm is capable of mitigating the influence of dishonest nodes by filtering out unfair ratings.

To test the proposed defence scheme further, we define three additional metrics: (a) *recognised proportion*, representing the number of dishonest recommendations identified by node i, (b) *false negative proportion*, indicating the number of dishonest recommendations identified as honest by node i,(c) *false positive proportion*, indicating the number of honest recommendations identified as dishonest by node i. Figs. 7 and 8 show the results for these three metrics in the presence of bad-mouthing and ballot-stuffing attack. The x-axis in Fig. 7a shows the percentage of bad-mouthing attack while y-axis shows the proportion of the recognised dishonest recommendation, false negative and false positive with the defence scheme in action. It can be observed that the defence algorithm can effectively mitigate the dishonest recommendation propagated by the bad-mouthing attackers regarding the recognition and false negative metrics. While it keeps the false positive proportion at a very low level (about 2 percent) when the attack percentage is more than 50 percent. Fig. 7b shows the case when the defence scheme is not in action. It can be seen that the proportion of recognised dishonest recommendation drops to less than 10 percent when the percentage of dishonest nodes increase to 80 percent and consequently the proportion of false negative increases with the increase in dishonest recommending nodes. As the defence scheme is not in action here, it accepts all the recommendations propagated in the network and updates the indirect trust value based on these recommendations. Therefore, the proportion of false positive remains at zero (Fig. 7b).

Fig. 8a shows results for ballot-stuffing attack. The proposed defence scheme is seen to be identifying dishonest recommendations and eliminating false negative effectively. The proportion of false positive is maintained at a reasonable level. The effect of dishonest recommendation in Fig. 8b is obvious. When there is no defence incorporated the proportion of recognition drops from about 0.9 to nearly 0.1 with variation of the ballot-stuffing attackers from 0.1 to 0.8. The false negative proportion also increases to nearly 0.9 with the increasing percentage of the dishonest recommending nodes.



(a) With Defence (b) Without Defence

Fig. 8. Recognised, false negative, and false positive proportion in the presence of ballot-stuffing attack for a) with defence; b) without defence.

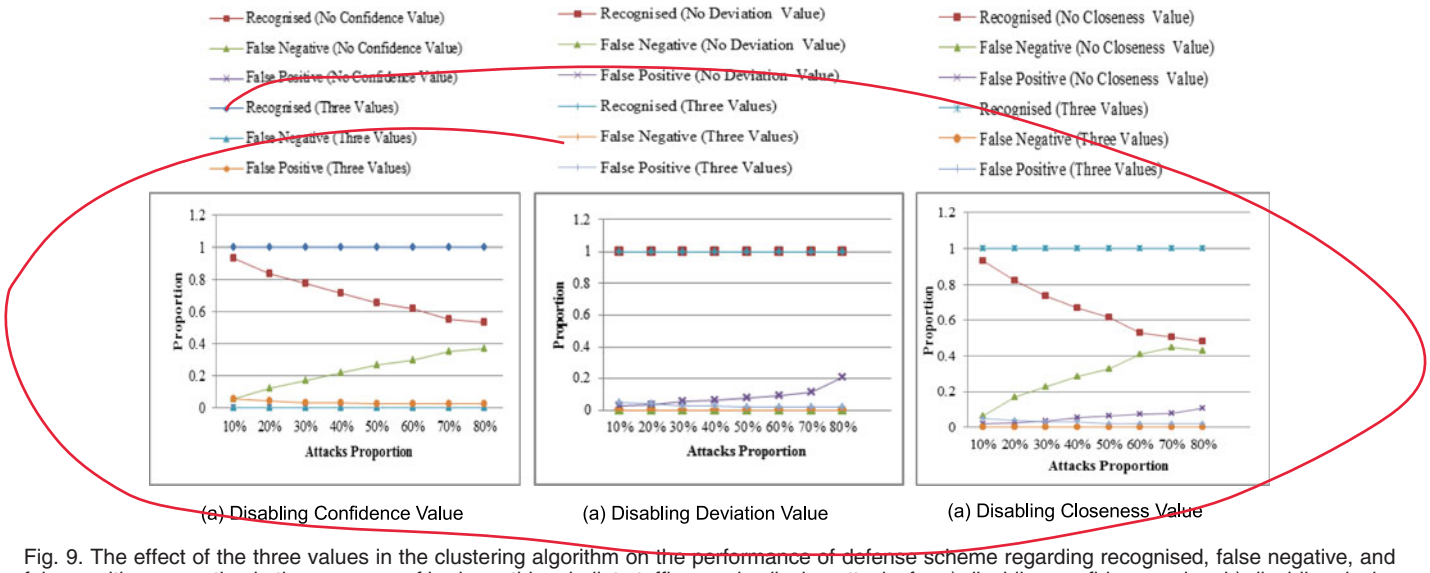


Fig. 9. The effect of the three values in the clustering algorithm on the performance of defense scheme regarding recognised, false negative, and false positive proportion in the presence of bad-mouthing, ballot-stuffing, and collusion attacks for a) disabling confidence value; b) disabling deviation value; disabling closeness value.

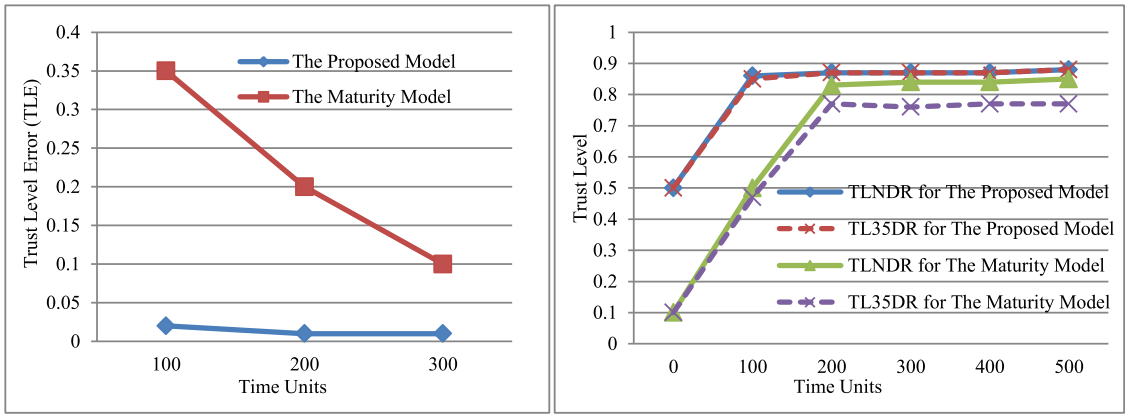
Furthermore, the proposed defence scheme is examined to observe the effect of each criterion (*recognised proportion*, *false negative proportion*, and *false positive proportion*) in clustering recommendations. The experiments are conducted over a range of various attack percentage. The different attacks considered are bad-mouthing, ballot-stuffing, and collusion. The results are shown in Fig. 9. First, the effect of the confidence value is tested by disabling it in the defence scheme and allowing the deviation and closeness value to work. It is obvious from Fig. 9a, that the defense scheme's performance is decreased in terms of recognised and false negative proportion of dishonest recommendation. The defense scheme is seen to be ineffective in recognising almost none of the dishonest recommendations propagated in the network by bad-mouthing, ballot-stuffing, and colluding attackers. On top, a number of false negative recommendations showed capable in penetrating the defense algorithm. The number of recognised proportion dropped with increase in the proportion of attack; from nearly 90 percent when just 10 percent of recommenders provide dishonest recommendations to nearly 50 percent when the dishonest recommenders reached 80 percent. On the other hand, false negative proportion increases with rise in the number of dishonest recommenders from very small proportion nearly 5 percent to nearly 40 percent at 80 percent of attack percentage. Interestingly, the number of false positive proportion is stable at 0 percent which means no honest recommenders were treated as dishonest. The reason being that the confidence value doesn't allow nodes without enough experiences to provide recommendation and this can result in treating some honest recommenders as dishonest. It can thus be concluded that the confidence value factor enhances the performance of the defense scheme by eliminating dishonest recommendations (even though it could result in a small proportion of false positive).

In second experiment, the deviation value is disabled in the clustering algorithm to understand its importance in the defense scheme. Fig. 9b shows that the performance of the defense scheme is reduced due to introduction of some false positive proportion in the case of disabling the deviation value. The proportion of false positive, which treats good nodes as dishonest has increased from 2 to over 20 percent when the proportion of attracters rises from 10 to 80 percent. The deviation value, however, has no significant effect on the performance of the recognised proportion of dishonest recommendations and the false negative proportion. The deviation value is still useful in the defense scheme as it has a strong relation with the confidence value during cold-start.

The third experiment tests the effect of closeness value. The experimental results, as shown in Fig. 9c, show that the closeness value has a strong impact on the three performance metrics of recognised, false negative and false positive proportions. With closeness value disabled, the proportion of recognised dishonest recommendations has fallen from 95 percent at 10 percent attact proportion to nearly 40 percent at 80 percent attact proportion. The results also show that the closeness value has a strong impact on the ability of the defense scheme in preventing the false negative. The number of false negative increases to more than 40 percent at 80 percent attack proportion when the closeness value is not used. Absence of the closeness value can also increase false positive—it has increasesed to nearly 10 percent when the dishonest attactors are 80 percent.

It can be seen from the experiment that the three proposed values have a varied level of positive impact on the performance of the defence scheme. Besides, the proposed values of the defense scheme are strongly correlated to work together in order to effectively prevent the influence of dishonest recommendations in the proposed trust models.

Finally, the performance of the proposed model is compared with the maturity model proposed in [23] in terms of two metrics: trust level error (TLE) which represents the proportion of error in evaluating the trust level



(a) Trust Level Error with Time Units by other nodes to evaluate node 8's Trust Level

(b) Good node 1's Trust Level with Time Units

Fig. 10. Comparative study with maturity model.

of a node i (node 8 in this case); and trust level evaluation of a good node (node 1 in this case) by another node j in the network. We follow the same network configuration and node selection which is provided in the maturity model (see [23] for details) to conduct this experiment. In this configuration, a high speed network is presented with high node mobility, which is different from our first configuration. This configuration of the test network allows us to show the effectiveness of the proposed scheme. Fig. 10 shows the results of this experiment. Fig. 10a displays the trust level error over the simulation time. It can be seen that the proposed model can keep the TLE smaller than the error reported by the maturity model.

The TLE in case of the proposed model is stable for the entire time of evaluation and converges to very small value nearly 0.01 towards the later phase. While for the maturity model, the TLE value is high initially (0.35) as compared to that of the proposed model and this only converged to 0.1 towards the end (time unit 3,000). Fig. 10b shows the effectiveness of the proposed defence scheme in evaluating the trust value of a good node (node 1) from the network. It considers the following scenarios: the expected trust value when there is no dishonest recommendation (TLNDR), and the same when there is 35 percent dishonest recommendation (TL35DR) both for the proposed model and the maturity model. The results show that the proposed model with the defence scheme can manage to avoid the dishonest recommendation and keep the trust value of node 1 near to the expected value and slightly higher than the results of the maturity model.

6.2 Cost of the Defence Scheme

Mobile ad-hoc networks are characterised by constrained resources in terms of communication, memory usage and computational complexity requirements. Any proposed model or defence scheme must reflect the trade-offs between accuracy of trustworthiness and network performance. As gathering and propagating trust information among distributed node can consume more resources of energy and time, it can enhance the decision making. Dynamic and highly mobile networks which suffer from several points of failure require techniques to enhance the decision making on nodes trustworthiness. However, the proposed defence scheme is lightweight in several aspects. In terms of communication, the proposed model is suitable for MANETs because only recommendation request and reply packets are used to send and receive a list of recommendations.

The packets of recommendations are exchanged between a single source of information which is represented in the recommendation manager to and from the evaluating node and the recommending nodes. The data size and length is very small as every recommending node provides just three parameters of accumulated positive and negative observations and its current position. The communication is also enhanced by on-demand scheme in which recommendation is requested whenever needed. Therefore, the defence scheme is conducted without network flooding and acquisition delay. The defence scheme is characterised with the advantage of a role-based management scheme for filtering dishonest recommendation in which three different components are interoperated to accomplish the task. The use of clustering in distributed networks can facilitate the data aggregation and reduce the computational power by each node to evaluate the trustworthiness of other nodes. One of the costs put on the proposed defence is the complexity that can be countered in maintaining the cluster and selecting the most trustworthy cluster. Another cost is the memory consumption in which the defence scheme consumes more memory to store recommendation for a period of time for conducting the filtering algorithm by the recommendation and clustering managers which is run by the evaluating node but no memory consumption on the side of the evaluated node. An additional cost is the time consumption which is more than the traditional defence which uses single recommender information to update the trustworthiness of the evaluated node. These costs can be reduced in the proposed defence scheme by using only the very last recommendations to be including in the clustering filtering computation. Dynamic selection of the number of recommendations based on a period of time can have many advantages, (1) reduce complexity and memory usage, (2) exclude any old recommendation from the calculation, (3) reduce the time that is used to select the trustworthy cluster.

7 Conclusion

A recommendation based trust model with a defence scheme is developed and analysed to filter attacks related to dishonest recommendation exchanged by nodes in the MANET. The use of recommendation can efficiently allow nodes to acquaint with each other without previous interactions but it exposes nodes to dishonest and unfair recommendation. Therefore, the proposed defence scheme utilises the clustering technique to filter out unfair recommendations exchanged by nodes in the network based on three values: (a) the level of confidence held by a node about others, (b) deviation threshold which ensures the unity of views between evaluating node and the evaluated node, and (c) closeness centrality value to ensure that recommending node is a close friend to the evaluating node for a period of time. The proposed model is tested by extensive simulation in terms of throughput and packet loss, against both bad-mouthing and ballot-stuffing attacks, and also compared with other proposal. The simulation results indicate that the proposed defence scheme can safely incorporate correct indirect trust evidences received by recommendations and eliminate untrustworthy ones. Moreover, it reduces the effect of false negative and false positive problems in selecting recommending nodes. The proposed model can be extended by weighting recommendations based on time and location to mitigate the influence of location and time dependent attacks.

References

[1] H. Deng, W. Li, and D. Agrawal, "Routing security in wireless ad hoc networks," *IEEE Commun. Mag.*, vol. 40, no. 10, pp. 70–75, Oct. 2002.

[2] B. Wu, J. Chen, J. Wu, and M. Cardei, "A survey of attacks and countermeasures in mobile ad hoc networks," in *Wireless Network Security*. New York, NY, USA: Springer, 2007, pp. 103–135.

[3] I. Chlamtac, M. Conti, and J. J.-N. Liu, "Mobile ad hoc networking: Imperatives and challenges," *Ad Hoc Netw.*, vol. 1, no. 1, pp. 13–64, 2003.

[4] W. Li, J. Parker, and A. Joshi, "Security through collaboration and trust in MANETs," *Mobile Netw. Appl.*, vol. 17, no. 3, pp. 342–352, 2012.

[5] J.-H. Cho, A. Swami, and R. Chen, "A survey on trust management for mobile ad hoc networks," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 562–583, 4th Quarter 2011.

[6] Z. Yan, P. Zhang, and T. Virtanen, "Trust evaluation based security solution in ad hoc networks," in *Proc. 7th Nordic Workshop Secure IT Syst.*, 2003, no. 14, pp. 1–10.

[7] A. A. Pirzada and C. McDonald, "Establishing trust in pure ad-hoc networks," in *Proc. 27th Australasian Conf. Comput. Sci.*, 2004, vol. 26, pp. 47–54.

[8] N. Pissinou, T. Ghosh, and K. Makki, "Collaborative trust-based secure routing in multihop ad hoc networks," in *Proc. Netw. Netw. Technol., Services, Protocols; Perform. Comput. Commun. Netw.; Mobile Wireless Commun.*, 2004, pp. 1446–1451.

[9] S. Buchegger and J. Y. Le Boudee, "Self-policing mobile ad hoc networks by reputation systems," *IEEE Commun. Mag.*, vol. 43, no. 7, pp. 101–107, Jul. 2005.

[10] R. Li, J. Li, P. Liu, and J. Kato, "A novel hybrid trust management framework for MANETs," in *Proc. 29th IEEE Int. Conf. Distrib. Comput. Syst. Workshops* , 2009, pp. 251–256.

[11] S. Ganeriwal, L. K. Balzano, and M. B. Srivastava, "Reputation-based framework for high integrity sensor networks," *ACM Trans. Sensor Netw.*, vol. 4, no. 3, p. 15, 2008.

[12] Y. Yu, K. Li, W. Zhou, and P. Li, "Trust mechanisms in wireless sensor networks: Attack analysis and countermeasures," *J. Netw. Comput. Appl.*, vol. 35, no. 3, pp. 867–880, 2012.

[13] C. Zouridaki, B. L. Mark, M. Hejmo, and R. K. Thomas, "E-Hermes: A robust cooperative trust establishment scheme for mobile ad hoc networks," *Ad Hoc Netw.*, vol. 7, no. 6, pp. 1156–1168, 2009.

[14] S. R. Zakhary and M. Radenkovic, "Reputation-based security protocol for MANETs in highly mobile disconnection-prone environments," in *Proc. 7th Int. Conf. Wireless On-Demand Netw. Syst. Services.*, 2010, pp. 161–167.

[15] H. Xia, Z. Jia, L. Ju, X. Li, and Y. Zhu, "A subjective trust management model with multiple decision factors for MANET based on AHP and fuzzy logic rules," in *Proc. IEEE/ACM Int. Conf. Green Comput. Commun.*, 2011, pp. 124–130.

[16] G. Soniand and K. Chandrawanshik, ''A novel defence scheme against selfish node attack in MANET," *Int. J. Comput. Sci. Appl.*, vol. 3, no. 3, pp. 51–63, 2013.

[17] S. Buchegger and J. Y. Le Boudec, "A robust reputation system for P2P and mobile ad hoc networks," in *Proc. 2nd Workshop Econ. P2P Syst.*, 2004, http://www.eecs.harvard.edu/p2pecon/program.html

[18] H. Li and M. Singhal, "Trust management in distributed systems," *Computer*, vol. 40, no. 2, pp. 45–53, Feb. 2007.

[19] P. Michiardi and R. Molva, "Core: A COllaborative REputation mechanism to enforce node cooperation in Mobile Ad Hoc Networks," in *Proc. Commun. Multimedia Security Conf.*, 2002, pp. 107–121.

[20] K. Wang, M. Wu, and S. Shen, "A trust evaluation method for node cooperation in mobile ad hoc networks," in *Proc. 5th Int. Conf. Inform. Technol.: New Generations*, 2008, pp. 1000–1005.

[21] J. Luo, X. Liu, and M. Fan, "A trust model based on fuzzy recommendation for mobile ad-hoc networks," *Comput. Netw.*, vol. 53, no. 14, pp. 2396–2407, 2009.

[22] S. Soltanali, S. Pirahesh, S. Niksefat, and M. Sabaei, "An efficient scheme to motivate cooperation in mobile ad hoc networks," in *Proc. 3rd Int. Conf. Netw. Services*, 2007, pp. 98–98.

[23] P. B. Velloso, R. P. Laufer, D. de O Cunha, O. C. M. Duarte, and G. Pujolle, "Trust management in mobile ad hoc networks using a scalable maturity-based model," *IEEE Trans. Netw. Service Manage.*, vol. 7, no. 3, pp. 172–185, Sep. 2010.

[24] H. Yu, S. Liu, A. C. Kot, C. Miao, and C. Leung, "Dynamic witness selection for trustworthy distributed cooperative sensing in cognitive radio networks," in *Proc. IEEE 13th Int. Conf. Commun. Technol.*, 2011, pp. 1–6.

[25] J. Cai, P. Yi, J. Chen, Z. Wang, and N. Liu, "An adaptive approach to detecting black and gray hole attacks in ad hoc network," in *Proc. 24th IEEE Int. Conf. Adv. Inform. Netw. Appl.*, 2010, pp. 775–780.

[26] K. Hoffman, D. Zage, and C. Nita-Rotaru, "A survey of attack and defense techniques for reputation systems," *ACM Comput. Surveys*, vol. 42, no. 1, p. 1, 2009.

[27] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Syst.*, vol. 43, no. 2, pp. 618–644, 2007.

[28] Y. Ma, H. Lu, and Z. Gan, "An improved direct trust evaluation algorithm for the context-aware trust model, " in *Proc. 7th Int. Conf. Innovative Mobile Internet Services Ubiquitous Comput.*, 2013, pp. 196–201.

[29] Y. Ren, G. Li, J. Zhang, and W. Zhou, "Lazy collaborative filtering for data sets with missing values," *IEEE Trans. Cybern.*, vol. 43, no. 6, pp. 1822–1834, Dec. 2013.

[30] G. Theodorakopoulos and J. S. Baras, "Trust evaluation in ad-hoc networks," in *Proc. 3rd ACM Workshop Wireless Security*, 2004, pp. 1–10.

[31] G. V. Crosby, L. Hesterand, and N. Pissinou, "Location-aware, trust-based detection and isolation of compromised nodes in wireless sensor networks," *Int. J. Netw. Security*, vol. 12, no. 2, pp. 107–117, 2011.

[32] M. K. Denko, T. Sun, and I. Woungang, "Trust management in ubiquitous computing: A Bayesian approach," *Comput. Commun.*, vol. 34, no. 3, pp. 398–406, 2011.

[33] E. M. Daly and M. Haahr, "Social network analysis for information flow in disconnected delay-tolerant MANETs," *IEEE Trans. Mobile Comput.*, vol. 8, no. 5, pp. 606–621, May 2009.

[34] T. Issariyakul and E. Hossain, *Introduction to Network Simulator NS2*. New York, NY, USA: Springer, 2011.

[35] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 2, no. 3, pp. 257–269, Jul.–Sep. 2003.

[36] H. Simaremare, A. Syarif, A. Abouaissa, R. F. Sari, and P. Lorenz, "Performance comparison of modified AODV in reference point group mobility and random waypoint mobility models," in *Proc. IEEE Int. Conf. Commun.*, 2013, pp. 3542–3546.
[37] J.-L. Huang and M.-S. Chen, "On the effect of group mobility to data replication in ad hoc networks," *IEEE Trans. Mobile Comput.*, vol. 5, no. 5, pp. 492–507, May 2006.

Antesar M. Shabut received the bachelor's degree in computer science from the University of Benghazi in 1999 and the master's degree in IT consultancy from the Sheffield Hallam University, United Kingdom, in 2008. She is currently working toward the PhD degree at the University of Bradford. Prior to this, she was a lecturer at the Faculty of Information Technology of Benghazi University of Libya from 2008. Her current research interests include trust and reputation management, security issues in mobile ad hoc networks and recommender systems.

Keshav P. Dahal received the MSc and PhD degrees from Starthclyde. He is a professor of Intelligent Systems in the Artificial intelligence, Visual Communications & Networks (AVCN) Research Centre at the University of the West of Scotland (UWS), United Kingdom. Prior to this, he was with the University of Bradford and the University of Strathclyde, United Kingdom. His research interests lie in the areas of applied AI to intelligent systems, trust and security modelling in distributed systems, and scheduling/optimization problems. He has published more than 95 peer-reviewed journal/conference papers, and has sat on organising/programme committees of more than 50 international conferences including a programme chair of SKIMA2006 and SKIMA2013. He is a senior member of the IEEE.

Sanat Kumar Bista did PhD research at the University of Bradford, United Kingdom, which involved modeling a game theoretic framework to assess the trustworthiness of players interacting in e-commerce and mobile ad-hoc network-like settings. He is a research fellow in ICT with the CSIRO, Australia. His research interest lies in the areas of trust, privacy, and security in settings such as social networks, e-commerce, mobile ad hoc networks, and the cloud. Prior to joining CSIRO, he was an assistant professor with the Department of Computer Science and Engineering at Kathmandu University. He has been on the program committee of numerous international conferences and is involved in several international scientific journals.

Irfan U. Awan received the PhD degree from the University of Bradford, United Kingdom, in 1997 and joined the Department of Computing at the same university in 1999 as a lecturer. He is a professor of computer science in the Department of Computing, University of Bradford, United Kingdom. He has published more than 200 papers in prestigious international journals and refereed conferences. He has edited several special issues of international journals and organised various international conferences. His research interests include network security, performance modellng, and evaluations of communication systems.